|  |  |
| --- | --- |
| Sl. No. | QUERY |
| 1. | SELECT  EXTRACT(year FROM OrderDate) AS Year,  SUM(Price \* Quantity) AS TotalSales  FROM  Data\_Sales  GROUP BY  Year  ORDER BY  TotalSales DESC  LIMIT 1; |
|  |  |
| 2. | ALTER TABLE Data\_Sales  ADD COLUMN Year INT,  ADD COLUMN Month INT,  ADD COLUMN Day INT;  UPDATE Data\_Sales  SET  Year = EXTRACT(year FROM OrderDate),  Month = EXTRACT(month FROM OrderDate),  Day = EXTRACT(day FROM OrderDate); |
|  |  |
| 3. | SELECT  SalesPerson,  LENGTH(SalesPerson) AS NameLength  FROM  Data\_Sales  GROUP BY  SalesPerson; |
|  |  |
| 4. | ALTER TABLE Data\_Persons  ADD COLUMN SalesManagerName VARCHAR(255);  UPDATE Data\_Persons  SET  SalesManagerName = LOWER(CONCAT(SalesPerson, '-', Manager)); |
|  |  |
| 5. | ALTER TABLE Data\_Persons  ADD COLUMN FirstName VARCHAR(255),  ADD COLUMN LastName VARCHAR(255);  UPDATE Data\_Persons  SET  FirstName = SPLIT\_PART(SalesPerson, ' ', 1),  LastName = SPLIT\_PART(SalesPerson, ' ', 2); |
|  |  |
| 6. | SELECT  CASE  WHEN data\_sales.producttype IN ('Product 1', 'Product 2', 'Product 3') THEN 'Category 1'  WHEN data\_sales.producttype IN ('Product 4', 'Product 5') THEN 'Category 2'  ELSE 'Other'  END AS Product\_Category,  SUM(Price \* Quantity) AS TotalRevenue  FROM  Data\_Sales  GROUP BY  Product\_Category  ORDER BY  TotalRevenue DESC; |
|  |  |
| 7. | CREATE TABLE sql\_assignment2.Orders (  "Row ID" INT PRIMARY KEY,  "Order ID" TEXT,  "Order Date" DATE,  "Ship Date" DATE,  "Ship Mode" TEXT,  "Customer ID" TEXT,  "Customer Name" TEXT,  Segment TEXT,  "Country/Region" TEXT,  City TEXT,  "State/Province" TEXT,  "Postal Code" TEXT,  Region TEXT,  "Product ID" TEXT,  Category TEXT,  "Sub-Category" TEXT,  "Product Name" TEXT,  Sales FLOAT,  Quantity INT,  Discount FLOAT,  Profit FLOAT  );  CREATE TABLE sql\_assignment2.People (  "Regional Manager" TEXT,  Region TEXT  );  CREATE TABLE sql\_assignment2.Returns (  Returned TEXT,  "Order ID" TEXT  ); |
|  |  |
| 8. | WITH RegionAvg AS (  SELECT  Region,  AVG(Price \* Quantity) AS AvgSalesRegion  FROM  Data\_Sales  GROUP BY  Region  )  SELECT  DS.\*,  RA.AvgSalesRegion  FROM  Data\_Sales DS  JOIN  RegionAvg RA ON DS.Region = RA.Region  WHERE  (DS.Price \* DS.Quantity) < RA.AvgSalesRegion; |
|  |  |
| 9. | WITH ClassifiedOrders AS (  SELECT  o.Sales,  CASE  WHEN r."Order ID" IS NOT NULL THEN 'Returned'  ELSE 'Non-Returned'  END AS Return\_Status  FROM sql\_assignment2.Orders o  LEFT JOIN sql\_assignment2.Returns r ON o."Order ID" = r."Order ID"  )  SELECT  Return\_Status,  AVG(Sales) AS Average\_Sales  FROM ClassifiedOrders  GROUP BY Return\_Status; |
|  |  |
| 10. | SELECT o."Order ID", o.Sales  FROM sql\_assignment2.Orders o  WHERE EXISTS (  SELECT 1  FROM sql\_assignment2.Returns r  WHERE o."Order ID" = r."Order ID"  ); |
|  |  |
| 11. | WITH RegionalSales AS (  SELECT  p."Regional Manager",  o.Sales  FROM sql\_assignment2.Orders o  JOIN sql\_assignment2.People p ON o.Region = p.Region  )  SELECT  "Regional Manager",  SUM(Sales) AS Total\_Sales  FROM RegionalSales  GROUP BY "Regional Manager"; |
|  |  |
| 12. | -- Step 1: Calculate total revenue for each Segment  WITH TotalRevenue AS (  SELECT  Segment,  SUM(Sales) AS Revenue  FROM sql\_assignment2.Orders  GROUP BY Segment  ),  -- Step 2: Calculate total number of customers for each Segment  TotalCustomers AS (  SELECT  Segment,  COUNT(DISTINCT "Customer ID") AS NumberOfCustomers  FROM sql\_assignment2.Orders  GROUP BY Segment  )  -- Step 3: Compute ARPU for each Segment  SELECT  r.Segment,  r.Revenue / c.NumberOfCustomers AS ARPU  FROM TotalRevenue r  JOIN TotalCustomers c ON r.Segment = c.Segment; |
|  |  |
| 13. | SELECT  "Order ID",  "Ship Mode",  Sales,  AVG(Sales) OVER (PARTITION BY "Ship Mode") AS Average\_Sales\_By\_Ship\_Mode,  MIN(Sales) OVER (PARTITION BY "Ship Mode") AS Min\_Sales\_By\_Ship\_Mode,  MAX(Sales) OVER (PARTITION BY "Ship Mode") AS Max\_Sales\_By\_Ship\_Mode  FROM sql\_assignment2.Orders; |
|  |  |
| 14. | WITH RankedOrders AS (  SELECT  Region,  "Order ID",  Sales,  ROW\_NUMBER() OVER (PARTITION BY Region ORDER BY Sales DESC) AS DescRank,  ROW\_NUMBER() OVER (PARTITION BY Region ORDER BY Sales ASC) AS AscRank  FROM sql\_assignment2.Orders  )  SELECT  Region,  "Order ID",  Sales  FROM RankedOrders  WHERE DescRank = 2 OR AscRank = 2; |
|  |  |
| 15. | WITH YearlySales AS (  SELECT  EXTRACT(YEAR FROM "Order Date")::INT AS SalesYear,  SUM(Sales) AS TotalSales  FROM sql\_assignment2.Orders  GROUP BY EXTRACT(YEAR FROM "Order Date")  ORDER BY SalesYear  )  SELECT  SalesYear,  TotalSales,  TotalSales - LAG(TotalSales, 1) OVER (ORDER BY SalesYear) AS YoY\_Sales\_Change  FROM YearlySales; |
|  |  |
| 16. | SELECT  o.Category,  SUM(o.Sales) AS Total\_Returned\_Sales  FROM sql\_assignment2.Orders o  JOIN sql\_assignment2.Returns r ON o."Order ID" = r."Order ID"  GROUP BY o.Category; |
|  |  |
| 17. | SELECT  o."Order ID",  o.Sales,  o.Category  FROM sql\_assignment2.Orders o  LEFT JOIN sql\_assignment2.Returns r ON o."Order ID" = r."Order ID"  WHERE r."Order ID" IS NULL; |
|  |  |
| 18. | SELECT  o."Order ID",  o.Sales,  o.Category  FROM sql\_assignment2.Orders o  INNER JOIN sql\_assignment2.Returns r ON o."Order ID" = r."Order ID"; |